AD-A031 892 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 9/2

MINI AND MICRO COMPUTERS IN COMMUNICATIONS.(U)

SEP 76 V E HIPKISS, C R SCHRAMM

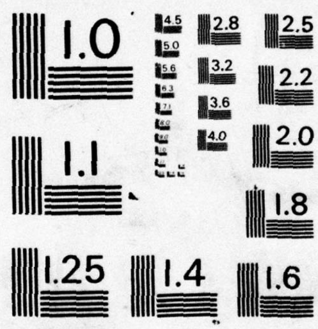
UNCLASSIFIED NL

1 OF 1

AD
A031892

END

DATE
FILMED
1-77



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MINI AND MICRO COMPUTERS IN
COMMUNICATIONS

by

Victor Edward Hipkiss
Carl Robert Schramm

September, 1976

Thesis Advisor; N. F. Schneidewind

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) Mini and Micro Computers in Communications.		5. TYPE OF REPORT & PERIOD COVERED (9) Master's Thesis, September 1976
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Victor Edward Hipkiss Carl Robert Schramm		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE (11) September, 1976
		13. NUMBER OF PAGES 73 (12) 72p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Mini and Micro Computers in Communications

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Mini and micro computers represent the newest developments in the computer field. Their small size, low cost, and wide ranging versatility make them valuable devices for use in communications systems. They can be used to perform many of the message handling functions previously performed by the main computer, thus freeing it for its more important application

DD FORM 1473 1 JAN 73 (Page 1) EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

251 450

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

bpg

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

tasks. The use of mini and micro computers as front end processors, data concentrators and terminal controllers is investigated, as well as general hardware and software features such small computers must have to perform communications functions. A cost comparison between minicomputers, microcomputers and hardwired devices is also presented.

ACCESSION for		
NTIS	White Section	☒
DDC	Buff Section	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or SPECIAL	
A		

DD Form 1473
1 Jan 73
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

MINI AND MICRO COMPUTERS IN COMMUNICATIONS

by

Victor E. Hipkiss
Lieutenant
Bachelor of Science

Carl R. Schramm
Lieutenant
Bachelor of Science

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 1976

Authors: Victor E Hipkiss
Carl R Schramm

Approved by: N. F. Schneidewind
Thesis Advisor

V. Michael Powers
Second Reader

Carl Jones
Chairman, Administrative Sciences Department

J. R. Borsting
Academic Dean

ABSTRACT

Mini and micro computers represent the newest developments in the computer field. Their small size, low cost, and wide ranging versatility make them valuable devices for use in communications systems. They can be used to perform many of the message handling functions previously performed by the main computer, thus freeing it for its more important application tasks. The use of mini and micro computers as front end processors, data concentrators and terminal controllers is investigated, as well as the general hardware and software features such small computers must have to perform communications functions. A cost comparison between minicomputers, microcomputers and hardwired devices is also presented.

TABLE OF CONTENTS

I. INTRODUCTION

Due to rapid technological advances within the past two decades, the computer industry has been dramatically reducing the size and cost of machines without a detrimental reduction of their capabilities. Today that trend continues due primarily to advances in integrated circuit construction techniques which have made possible large scale integration (LSI). LSI techniques have made possible the construction of microprocessors condidered by some to be potentially one of the most significant and important developments in the computer field since its advent. The microprocessor in turn has made possible the the so called microcomputer.

These new smaller computers have been adapted to many fields ranging from auto traffic control to monitoring delicate instruments in a laboratory environment. This paper will address just one of the areas where mini and micro computers have been making a large impact, namely, the telecommunications field. First, a brief history of mini and micro computers will be given in order to present the background and development of these small computers and to show their general progress to date. Next the general characteristics of mini and micro computers will be discussed; specifically mentioned will be their hardware, software, and capabilities. Following this discussion, the rational for using mini and micro computers in communications systems will be presented. Why they are needed, where they are used, and the necessary characteristics these computers must have for communications applications will be discussed. Finally, examples of present systems will be discussed.

II. HISTORY

The evolution of computer circuit technology over the past twenty years has advanced from vacuum tubes as the primary electronic device through transistors to the present day integrated circuits. It is now about fifteen years since the electronics industry learned to make miniature electronic circuits on a chip of silicon substrate by alternating processes of masked etching and diffusion. This made possible the development of the minicomputer, the first embodiment of which was the PDP-5 and PDP-8, small parallel data processors introduced in 1963 and 1965 by the Digital Equipment Corporation. In the early 1960's the commercially available integrated circuits incorporated at most a score of components such as diodes, transistors and resistors. Production yields (the fraction of circuits that worked) were low, and packaging technology did not allow the realization of practical devices with more then a dozen leads, or connections. The basic technology, however, was so amenable to improvement and the rivalry among manufacturers was so keen that every year since then the number of components that could be econcomically placed on a single chip has doubled. Today chips less then a quarter of an inch on an edge can incorporate well over 20,000 components. As a result, the cost per component has in 10 years dropped by a factor of more then 100, from about 20 cents to a small fraction of a cent [1].

The PDP-8 was approximately 34"x20"x22" yet nevertheless was at least a minimal computer. Thus it was physical size that gave rise to the term "minicomputer".

Within a decade the PDP-8 had been widely imitated and had given birth to an entire industry concerned not only with minicomputers themselves, but also with software and innumerable peripheral devices and auxiliary services.

The minicomputer industry has been successful because of the dramatic advances in microelectronics that took place throughout the 1960's. These advances have led to steadily increasing performance and steadily decreasing cost and size (about 30 percent per year). In 1976 minicomputers can be found in existing systems in ever increasing numbers. They bring with them cost reductions and the flexibility and simplified design of stored program machines, and have made possible a wide range of new applications that call for an inexpensive resident computer.

With the continuing advances in microelectronics it was felt by some to be just a question of time until the further integration of microscopic components would lead to a microcomputer, a machine that would have tens of thousands of components on a single chip, or at most a few chips, and that would require no more than a few hundred milliwatts of power.

By the late 1960's it was still difficult to predict when single-chip computers would become a reality because semiconductor manufactures had been concentrating on the development of what are termed bipolar devices. In such devices the currents in the transistors are carried by both electrons and holes. With bipolar devices high component densities were difficult to achieve, and substrate sizes were limited because of low production yields and problems with dissipation of heat. At about this time, however, older fabrication problems that had plagued the metal-oxide-semiconductor (MOS) technology were solved, and it became economically possible to manufacture large

unipolar devices with high component densities and low heat production. Intially these devices were exclusively of the p-channel type (PMOS) [2].

The PMOS technology contributed to a major conceptual advance in 1971, when the Intel Corporation, which had undertaken to develop a calculator chip, chose to design it as a more versatile, programable, single chip microprocessor. A microprocessor is analogous to a central processing unit of a large computer. With the inclusion of a control memory and a master clock the microcomputer was born.

Microcomputers lie somewhere between microprocessors and minicomputers and present an alternative to each. This definition does not imply that the microcomputer is enclosed in a cabinet with switches and lights as does the normal mini and large-scale computers, although some of the currently available microcomputers do come that way. Futhermore, the microcomputer does not require an integral power supply, largely because the microcomputer may be part of a larger system, such as a numerical controller of machine tools or a traffic intersection controller, which itself will contain power supplies for all subsystems.

The microcomputer may be produced either by the microprocessor manufacturer, or by a company not involved with semi-conductor manufacturing.

As more engineers became aware of microprocessor technology, microprocessors were incorporated in a wider range of devices, and more semiconducter manufacturers fought to get a foothold in this new and expanding market.

Although there have been a few failures, more then 30 different microprocessors have appeared within the five

years since the introduction of Intel's first microprocessor. The older machines still use the PMOS technology, which has been developed to such a level that one 12-Bit microprocessor made in Japan by Toshiba, has 11,000 transistors on a single .22 inch by .24 inch chip. The newer microprocessors, however, use either the faster NMOS technology, in which the transistors operate by means of negative current carriers (electrons), or the complementary MOS (CMOS) technology. CMOS combines the PMOS and NMOS technologies to achieve a reduction in power requirements and to improve resistance to extraneous noise.

Microprocessor and microcomputer development came three decades after the first electronic computers and therefore benefited from the many years of experience accumulated in system organization and computer architecture. Frequently, many advanced concepts and features not available in larger more expensive computers are routinely included in microcomputers. This is because advancing technology has made it possible for these features and concepts to be easily included in the newer microcomputers.

With the fast parallel development of many microcomputer systems, there has arisen a number of original designs and architectures. Thus the mini and micro computer can be used in many diverse fields.

In the following chapters a detailed analysis of mini and micro computers and their capabilities in the communication field will be presented.

III. MINICOMPUTERS: GENERAL CHARACTERISTICS

In this section the general characteristics of the minicomputer will be discussed. It must be realized that in the computer industry if the question was asked of experts: "What is a minicomputer?"; the answers would vary depending on who was asked. The reason for this ambiguity is the rapid development of technology in the industry, and the constant changing of cost, size, and capability of equipment. Much of the terminology found in the computer industry does not have a hard and fast definition. This is also true of minicomputers.

To give an example of how definitions can vary, the Rand Report [3] describes minicomputers as costing from $4,000 to $20,000 for a processor and 4K of memory [4]. DATAMATION describes them as selling for under $10,000 in a useable configuration with 4K of memory; DATAPRO REPORTS states a minicomputer costs below $100,000 [5].

The typical minicomputer, then, is a parallel binary processor with a 16-Bit word length (though 8-Bit, 12-Bit, and 24-Bit word lengths are also common). It employs integrated circuits and is housed in a compact cabinet suitable for either tabletop use or mounting in a standard 19-inch rack. It weighs less then 50 pounds, consumes less then 500 watts of standard 115-volt electrical power, and requires no special air conditioning nor facilities requirements. It offers from 4,069 to 65,536 words of magnetic core or semiconductor storage with a cycle time of 0.8 to 1.5 microseconds. Parity checking and storage protection are normally available as extra-cost options [5].

Today's typical minicomputer uses a one address instruction format and has two accumulators, a single index register, and a multi-level indirect addressing facility. The add time for 16-Bit operands is one to three microseconds. Hardware multiply/divide instructions are optional, as are power-failure protection and a real-time clock or timer. Floating-point arithmetic requires the use of software subroutines but in some cases is available through optional hardware

An optional direct memory access (DMA) channel, which accommodates I/O data rates of up to about 1,000,000 words per second, facilitates input/output operations in the typical minicomputer. Additionally, minicomputer manufacturers offer the widest variety of I/O bus arrangements in the EDP industry. Together with multiplexer and selector channels, a typical minicomputer has bus repeaters, unique bus structures, and the DMA channel, all of which provide direct access to memory from peripheral devices [5].

The typical complement of peripheral equipment comprises a teleprinter, disk storage unit, magnetic tape drive, card reader, paper tape reader and punch, line printer, and an assortment of interfaces for communication and control applications.

Although the above is a general discription of typical minicomputers on the market, each mini has a fundamental architectural difference reflecting the vendor's unique approach to solving a particular class of problem. It is not uncommon that a particular minicomputer is strong in some fuctional areas, while very weak in others. All, in fact, seem to have good and bad features.

For this reason it is very important that a careful match be made between particular minicomputers and the applications for which they will be used.

One of the most important features of minicomputers is their versatility.

Most minicomputers presently being manufactured are composed of chassis and power supply, a common bus structure connecting a number of card slots in that chassis, and a number of functionally independent circuit boards comprising the processor, memory, input/output (I/O) drives, and I/O channels. Each card is often functionally independent and transparent to the operation of the other cards such that adding or deleting a card does not affect the operation of the rest of the system. Further, each module may operate asynchronously of every other module, permitting each to operate at its maximum rated speed.

Thus one finds that additional memory or I/O drivers may be added simply by removing the back panel and plugging in additional circuit cards, and even processors may be changed or upgraded by removing one card and inserting another in its place. This allows considerable versatility for configuration changes and permits easy system upgrade to faster memories and processors as processing requirements increase [6].

Generally the instruction set for minicomputers is based on the system designer's idea of what the basic capabilities of the machine will be. But if the machine is ultimately put to use in some application the designer did not foresee, it may be necessary to perform a short sequence of instructions to simulate the desired instruction. This often results in no great deficiency except in the case where an iterative set of instructions must be performed to

simulate the desired instruction, such as adding to simulate multiplication. This repetitive addition would certainly be a deficiency if many multiplications needed to be performed.

One must also be wary of the total number of instructions a vendor states that a machine can perform. What is important is the number of useful instructions that the machine can perform <u>on an operand</u>. Vendors have a tendancy to count all instructions performed, even when some of them are not useful.

As stated earlier, most minicomputers can perform binary addition and subtraction, but frequently multiplication and division cannot be performed, or is an optional extra.

Many minicomputers also have no storage protection, but with the increased introduction of these machines into time sharing systems, the development of block designation schemes and memory mapping schemes is progressing. Also, it must be noted that, generally, memories in minicomputers are of the semiconductor type and require a constant biasing voltage to maintain their contents. This means that memory is lost when power is lost. Generally minicomputer manufactures provide an emergency battery power source in their systems to protect the memory during power failures [7].

There are many peripheral devices on the market that can be connected to minicomputers. The problem is that many of these devices were designed for large computer systems and are priced accordingly. As soon as a minicomputer is used as a general data processor and not for a specialized function, the cost of the peripheral systems soon outstrip the cost of the minicomputer itself.

Software has been a problem for large computer systems

in recent years; this is also true with minicomputers. A story in an IEEE SPECTRUM [8] article states: "Back in 1971, Charles Jackson, a software oriented type EE, visited a minicomputer manufacture's demonstration facility to try its machine. A salesman had assured him that the manufacturer's Fortran was working, so Mr Jackson tried to run a well debugged bench-mark program. It compiled, began to run, although giving strange answers, and finally halted abnormally. No one could explain this mysterious action." This example shows some of the problems that can be encountered with software.

This type of experience in the early days of minicomputers led many to say that software for minis was non-existent. This lack of software can be explained when we reflect upon how the minicomputer developed. When first introduced it was used mainly by industry and educational organizations which had the resources to program these cheap machines. As applications grew, and with a competitive market, the vendors had to provide some software, but it had to be minimal in order to keep the prices of the machines down to a competitive level.

One major trend has been towards more sophisticated systems software, the kind of software that manages computer system resources so that the user programs can operate efficiently. For example, tremendous progress is being made in memory management, with current minicomputer features often superior to those offered by large computers only a few years ago. The days are gone when a user had to load his own program in absolutely fixed, defined minicomputer memory areas. Today, minicomputer systems provide the user with automatic management of drums and discs, as well as main memory, so he no longer has to keep track of physical locations [8].

A key achievement of minicomputer systems software has been the use of input/output handlers: standard software packages that connect standard peripheral equipment into the system.

For example, when the user's program says GET and PUT data, the I/O handler for the punched card reader signals the reader to move a card, then it converts the card information into the format that the computer needs, and finally it transfers the information into the computer's memory.

A second trend is towards increased compatibility. Minicomputer manufacturers are becoming more conscious of the need to make their current software compatible with the software they introduced last year and plan to make available next year.

Hardware compatibility is already available for several lines of minicomputers; that is, there are families of machines that can use the same peripherals, have the same standard interfaces, memory controls, etc., and compatible software for these machines, with the flexibility of use that is available in conventional large computer systems.

Software as well as hardware compatibility is very important to the system designer who uses the minicomputers, because it allows him to start with a minimum system, test a process he wants to use, and then apply directly what he has learned in a full-scale system, without having to reprogram [9].

The third trend is towards greater accuracy for minicomputer software. Computational accuracy depends both on hardware and on software. For example, accurate floating point arithmetic can be implemented in software on an

inexpensive minicomputer, but from the minicomputer manufacture's viewpoint, writing these routines may require many man-months of programing effort, and the routines may be slow running. Using a more expensive minicomputer, capable of handling larger numbers during each memory cycle, the same accuracy might be obtained with less programing effort and faster run times. With floating point as a built-in hardware feature of the machine, the run times would still be faster while the programing effort would be minimal. Of course, such hardware adds to the cost of the computer system.

As minicomputers systems software becomes more like that of large computers, price advantages are moving minicomputers into areas traditionally reserved for large computers.

Minicomputer vendors are now offering software packages in many application areas. One example of a software package offered by General Automation in the area of telecommunications is the General Automation Telecommunication Access Method (GATAM-16). This package can transmit either EBCDIC, USASCII, or BCD and includes the following features [10]

- Bidding for a line to resolve contention
- Polling and selection in a multipoint environment
- Error checking
- Message block format
- Time out control
- Interpetation of data link control characters
- Synchronization
- Sequential checking of message blocks
- End of message transmission

The package runs on several of their minicomputers and also includes documentation and reference material. This package is included with their appropriate machine for a $2000 license fee.

From their first appearance on the market, minicomputers and their software have developed rapidly. It appears, however, that software will continue to be a problem.

IV. MICROCOMPUTERS: GENERAL CHARACTERISTICS

A. ARCHITECTURE

The distinguishing characteristic of a microcomputer is the microprocessor. A microprocessor is defined as one or more large-scale (LSI) chips that perform the basic functions of a processing unit, namely, accept data from an input device such as a keyboard or sensor, modify the data with arithmetic or logical functions, and output the data to some external device [11,12]. Some microprocessors also offer many internal registers, microprogrammed control units, simple I/O interfacing, and a common I/O address/data bus. A microprocessor combined with memory and I/O devices is a microcomputer. Typically, one chip is devoted to each of the functional elements of a processor; i.e., the central processor comprises one chip, as does the arithmetic logic unit, and so on [11].

Thus, a microprocessor has the usual standard elements of any computer processor. Its typical size of 0.16 inch on a side leads to the "micro" designation.

Present microcomputers are manufactured by means of metal oxide semiconductor (MOS) processing technologies. The technology represented by MOS offers extremely high-density transistors per package, but its speed is inherently slower than that of bipolar devices. Current MOS switching speeds for a logic element or chip range from 40 nanoseconds [11] for the potentially faster n-channel

silicon gate devices to 200 nanoseconds [11] for p-channel metal units. To compensate for the slower speed, architectural attributes which exploit MOS technology - namely, hardware index registers, highly parallel bus structures, register stacks with programmable stack pointers, and decimal arithmetic - have been added to increase the speed of microcomputers vis-a-vis bipolar devices.

The power consumption of LSI microprocessor is about 66 to 75 percent less than that of comparable minicomputer CPU's [13]. In general, system reliability is enhanced when power consumption is reduced. Of course, for a system containing a single CPU, the differences would not be significant considering the overall system power requirements. However, in applications requiring many CPU's, these differences can be substantial.

Currently, a MOS LSI microcomputer typically operates at about 33 to 50 percent of the speed of commercially available minicomputers. For example, typical memory-to-memory add times vary between 5 and 20 microseconds for moderately priced minicomputers, compared to 15 to 60 microseconds for microcomputers [13].

The overall architecture of most microprocessors is remarkably similar to that of bus-oriented minicomputer systems. The uses for microprocessors can generally be characterized by bit width. For instance, 4-bit chips are often used for decimal digit operations, whereas communication terminals use 8-bit words for character transmission codes [14].

The range of characteristics among the currently available microprocessors is broad [13]:

- Data word size - 4 to 100 bits
- Instruction set - 25 to 120 instructions
- Instruction format - 8 to 24 bits
- ROM (read-only memory) - 100 24-bit to 65K 8-bit words
- RAM (random-access memory) - up to 65K 16-bit words
- General-purpose registers - 1 to 16
- Time to fetch and execute one instruction 0.54 to 62 microseconds, with 5 to 10 microseconds common
- Stack depth - 2 to 32 levels
- Interrupt capability - none to full

Functionally, the microprocessor chip includes the arithmetic logic unit, the general purpose registers and the control-bus structure. The architecture consists of the partitioning of the processor between one or more chips, the number of pins each chip has, the chip size, the off-chip memory, and the I/O address and control bus structure [15].

Speed or throughput is very dependent on architecture. Microprocessor chips are relatively slow because of the number of pins (and hence the number of I/O lines) does not allow much parallelism. Therefore, more encoding is done and thus off-the-chip decoding is necessary. Clock speed (or frequency) is not necessarily indicative of execution speed. Speed is a function of data and address path widths, number of separate paths and overlap in the fetch and execute cycles [15].

Arithmetic and register operations in a microprocessor have evolved such that there is capability for both decimal and binary arithmetic. Because of the I/O limitations (or pin limitations, which is a bandwidth problem) from the chip to main memory, most architectures use a pushdown stack of

some sort. The stack helps the programmer minimize register transfers, facilitates counting and sorting, and limits needless transfers to and from main memory [15].

The memory section of a microcomputer usually accounts for a major portion of the chips. Three kinds of memory are used. Random access memory chips are used primarily for variable data and scratch pad. Read-only memory chips are used to store instruction sequences, and constant data. Programmable Read-Only Memory (PROM) chips are used for tailoring the general purpose microcomputers for specific applications [15].

RAMS are expensive compared to ROMS, but the data in ROMS must be stored at the time they are created, so there is a production delay associated with them as well as a "programming" cost. PROM chips, some of which can be erased by ultraviolet light and reprogrammed, are used in small quantities during the design stage to make logic changes. Once the design is firm, the lower cost ROM is produced in large quantities [15].

B. PERIPHERALS

At the high end of the microcomputer line, the minicomputer analogs, a wide range of peripherals and their interfaces are provided. Teletype, paper tape, and analog interface systems are standard peripheral devices for micros; there have recently been a few announcements of mass storage systems for micros, primarily floppy disk and cassette tape units, as these are the only candidates in a price range consistent with micros [16].

C. SOFTWARE

If a microcomputer is to be installed into a dedicated system where the memory requirements should be kept to a minimum to reduce cost, it is not necessary for the microcomputer to contain enough memory to run an assembler or compiler. If the microcomputer is to be used as a general purpose computer, however, then it is almost necessary for the assembler or compiler to run the system. Some microcomputers have this facility when provided with enough memory [17]. However, a compiler is frequently used to assemble a program on a large machine (one which has the memory size required for assembly) for execution on the microcomputer.

For a dedicated application, there should be a development system which has at least three software devices: (1) a compiler so programmers need not work in machine code, and (2) a debug package (run on the microcomputer) and (3) simulation software (run on another computer) to test the programs. To take advantage of a microcomputer's independence from other parts of the total system, the user should prefer to use some kind of debug program (perhaps combined with special hardware) rather then a simulator; the testing is much more valid when actually run on the microcomputer itself [17].

For larger, more general applications, a larger minimal operating system is a requirement. This must be at least a binary load/dump routine to read and save programs on some kind of permanent storage medium. As programs move beyond the trivial stage, a text editor becomes a big time saver, and if they get longer still, a compact higher level language is useful. Currently there are four major types of higher level languages available (Process Control Language,

versions of Basic, PLM and its derivatives, FORTRAN), with only two of them really meeting their design objective [18] (Process Control Language and MITS Basic).

V. APPLICATIONS

A. THE PROBLEM

The most rapidly growing use of the world's telecommunication links is for data transmission between computers. The reason is the power and versatility that the interlinking of computers can bring, plus the potential benefits to the individual of having this power at his fingertips.

The typical data processing system of a few years ago consisted of a large general purpose computer surrounded by a battery of peripheral devices for input and output of the data to be processed. As applications grew in sophistication the desire to enter and retrieve data at a remote location was fulfilled by adding another peripheral device that interfaced with a telephone line and provided access to a remotely located terminal. This peripheral device, commonly called a line adapter unit, provided a simple interface for the buffering of data characters between the data processing computer and the telephone line. However, it was completely dependent upon the processing capabilities of the host computer. Today applications are much more sophisticated. Applications involving hundreds of terminals and a wide variety of sources are commonplace. The simple line adapter unit has grown to become a very complex and a very expensive device [19].

The problem facing the system designer, as increasing

numbers of terminals are used with one computer system, is the need to devise means of lowering the overall cost as the network grows. Data transmission lines today work with various devices that were not found on early telegraph lines. The variety and scope of such devices is growing rapidly, and the system analyst is faced with a confusing array of different machines and claims. The purpose of much of the increased complexity is to reduce the overall network cost. The larger real-time communication networks of today would have been too expensive without concentrators, multiplexers, and without elaborate time-control procedures. On such systems one can no longer simply connect each terminal to a voice line, leased or dial up, to the computer. Furthermore, in many of today's applications a saturation point has been reached where the data processing computer is becoming seriously impaired by the communications processing function. Although the communications processing requirements are normally simple in terms of data manipulation, they are time-critical factors. This characteristic of being time-critical presents a difficult challenge to the large data processing computer. Typically, a large computer system is designed to work best when it can function continuously, executing a full set of program instructions on a given application before branching to another. Even the most modern machines now being produced suffer in efficiency if unscheduled interrupts requiring immediate service occur outside the control and expectation of the processor. This loss of efficiency and increased processing load has become such a serious problem that new approaches are needed to again develop balanced operations [20].

Before suggesting a new approach, however, it is first necessary to describe some of the methods used today to reduce communication cost. It will then be shown how mini/microcomputers can be applied to replace these

traditional devices and enhance the overall network cost and efficiency.

B. COST REDUCTION METHODS IN COMMUNICATIONS

1. Private Exchange

Private exchanges are used to reduce the total mileage from the different terminals to the computer.

In a private exchange a circuit switch connection is made between terminal and computer by means of a private switch unit. The advantage of this is that the user has control over the equipment used and the circuit quality. The major disadvantage of this system is that a terminal may not be able to connect with the computer because all lines are in use and therefore must wait for a free line. Also one terminal user often has the capability of hanging on to his connection. This, of course, degrades the whole system and ties up lines unnecessarily.

2. Multidrop Lines

The total line mileage can be cut if several terminals are connected to the same line. The terminals can be in different locations. In many cases the multidrop line can be designed to take the shortest path between terminal locations. Two terminals on a multidrop line cannot transmit or receive at the same time. A discipline must be established on the line whereby the devices wait their turn to transmit.

3. Multiplexers

These devices take one line and divide the transmission capacity into several different channels. There are two methods used to accomplish this; frequency division multiplexing (FDM) and time division multiplexing (TDM). The advantage to multiplexing of transmission is that several lower-rated lines can be combined into a larger-rated line and overall line costs can be reduced.

4. Buffers

Most transmitting and receiving devices operate at their own rated speed- a speed that is usually different from the optimum speed of the line they are connected to. An extreme example is a keyboard operator. The operator at the machine presses the keys at his own rate, and the characters are transmitted at this rate regardless of the capacity of the line. Buffers, on the other hand, can store the characters in a compact space and when the message is completed it can be transmitted at the optimum rate. Paper tape is one form of a cheap buffering system.

5. Terminal Control Units

Terminal control units are located in the vicinity of several terminals. Its function is to buffer the terminals and perform line control functions so that the optimum rate of data transmission from the terminals can be maintained.

6. Concentrators

Synchronous transmission is more efficient than start-stop transmission. A concentrator performs the function of combining several low speed lines on to a high speed line. The concentrator buffers the messages and then retransmits them in blocks at the higher speed. Since they hold messages and then retransmit them they are sometimes called "hold-and-forward" concentrators.

7. Message-Switching Systems

In message-switching systems there are many locations with one central switching center. The switching center stores the message from an incoming line and then forwards the message on the correct outgoing line so that the message will reach the correct destination.

8. Packet-Switching Systems

In this type of system messages are broken down into a predetermined size or packet. These packets are then routed over the quickest route to the correct destination. At the destination the packets are reassembled into the original message.

Keep in mind that while the above methods do indeed reduce communication costs when used properly, they can result in increased communications processing requirements for the central computer thus causing the problems decribed in the previous section. As mentioned earlier a new approch is needed.

C. THE SOLUTION

The approach that provides the best answer now and will continue to support the requirements in the future is one of a segregated data-communications function. Greater segregation is achieved by moving the interface between the computer system and the communications system from the data modem back into the I/O channels or memory ports of the data processing system. A system of communications controllers (or processors) can then be delegated the responsibility for such communication processing tasks as message accumulation, line discipline, error detection and correction, message formatting, message addressing and routing, message storing and forwarding, terminal polling, terminal control, data concentration and transmission speed conversion [21].

This system of communications controllers has the primary purpose of accomplishing the communication of data from one place to another in accordance with the needs of the data processing systems. Multiple use of data communication circuits and a totally integrated approach to the communication function can provide significant capability at moderate cost. Using the totally integrated approch the main computer does only application job processing, and any attached communications controllers perform only communications functions.

There are essentially three basic types of communications controllers that can be used as the building blocks of almost all present and future data communication systems. They are:

- communications front-end processors
- terminal controllers
- data concentrators

By using combinations of these basic components,

multipurpose communication networks are being established to furnish the data communication needs of entire companies and even entire industries. In addition, even though particular configurations are necessarily directed by specific applications, the system design can be very much independent of the end-use of the data.

Communications controllers configured as front-ends can be utilized at major data processing centers to act as the interfaces between the data processors and the communications network. In addition, these same controllers with software configured as message switching systems can link together the various elements of the communications network, passing messages from point to point.

Terminal controllers can be distributed throughout the network to interface with a variety of terminals as required by any particular teleprocessing application.

Data concentrator systems can combine the data traffic from many different sources on single communications paths. This helps to more efficiently utilize existing communications paths, especially the paths of the digital data communications networks.

The result is a more flexible approach to the data communication function then is possible when the main computer must do all the communications processing itself [21].

D. ROLE OF THE MINICOMPUTER

1. The Requirements of a Communications Minicomputer

It is the minicomputer that has made this new approach possible. A minicomputer surrounded by the proper interface hardware and a minimum of peripherals can become an efficient programable communications controller at a fraction of the cost of equivalent processing capability in a large computer [22]. Proper application of such a communications controller can add a significant measure of flexibility to the communications aspect of a network, thus further enhancing the potential cost savings of any system.

The minicomputer is particularly well adapted to the tasks associated with communications processing. The primary requirements for a communications processor are the ability to change rapidly between program states, to process data using logical operations, to randomly access entries in core resident lists and tables, and to conviently store and move character data. The nominally available 1 microsecond cycle time of the minicomputer is adequate for almost all applications. Memory requirements are primarily dependent upon the expected traffic throughput, the maximum amount of time a message remains resident in the computer, the complexity of the communications line discipline, the amount of message processing required and the number of system control tasks to be accomplished. Applications in which terminal control is a primary function will involve many task routines such as error detection and correction, message formatting, message addressing and routing, and so on. Thus large blocks of core, or a disc storage unit from which blocks can be retrieved when required, is also necessary [22].

The medium size random access disc is a very practical peripheral for a minicomputer being used in a communications application; not only can it provide storage for less frequently used task programs but it can easily be

used to store message data and large arrays of tabular information. The additional complexity of the software required to support a disc is usually justified by the added utility gained in the system.

Only a few specific features make one computer more desirable than another. The data to be handled and stored by the computer is normally coded alphanumeric data. All standard codes are usually 8 data bits or less, therefore a standard 16 bit memory word can easily accommodate two characters. Although a 12 bit word is adequate for two five bit BCD characters, it is not sufficient for two ASCII characters. Since it is desirable to have two characters of data stored per memory word, the computer selected should have half-word or byte manipulation.

The inclusion of extensive hardware arithmetic routines for fast multiply and divide or double precision is not required since they are seldom used in communications applications. Much more important is an extensive array of logical instructions for manipulating and testing data. A significant amount of table manipulation and mask testing can be expected in communications applications. Therefore, instructions that assist in the preparation of efficient table search routines and data movement routines are extremely important.

One of the primary requirements of the communications minicomputer is for it to be capable of high data rates for transferring data into and out of data storage locations. Therefore, a convenient I/O path is required. Direct memory access channels are usually a requirement. Minicomputers that transfer each character through a register as a result of the execution of a series of I/O commands are too inefficient for a controller application [22].

2. The Advantage of a Front End Processor

If a large computer is used for interactive processing then, as a minimum, such functions as monitoring the status of transmission lines, bit detection, character assembly and message assembly have to be carried out. Messages will then have to be stored for processing later. The large computer would carry out these functions during the processing of other jobs in the system. However, it is by no means true that the large expensive computers must do this. A lot of line handling can be time consuming and tedious, especially for the large computer. Moreover, for many of the other message assembly functions, it seems rather foolish to use 32-bit words of which, frequently, only one bit at a time is processed [23].

As a result, an ever-increasing number of minicomputers are employed as front ends intended to relieve a larger digital computer of input/output routines. A data channel designed to implement block transfers of input and output data (and to perform a few extra device-control chores) acquires many of the features of a small digital processor. A minicomputer transferring data blocks by direct memory access and communicating with a larger digital computer through programmed instructions and interrupts can perform data-transfer and control operations equivalent to those of several data channels; like a data channel, it accepts programmed instructions to preset address counters, word counters, and control registers and, in turn, communicates to the larger digital computer through processor interrupts [24].

But a minicomputer can do much more than transfer data blocks and control device functions. The minicomputer

memory buffers external devices, which can thus operate at their optimal speed without waiting for the main digital computer program, and vice versa. In addition if there is time, the minicomputer peripheral processor can perform formatting, scaling, and code-changing operations, sense and signal error conditions, and perform parity and syntax checks. What is more, many input/output programs, interrupt service routines, etc. can be stored in minicomputer memory at substantially lower cost than is possible in the more expensive large-computer memory with its greater word length. Tens of thousands of bytes of main-computer core storage may be saved in this manner. Minicomputers have been used as peripheral processors for practically all types of peripherals, such as multiple tape units, and communication interfaces. Such applications favor minicomputer instruction sets which permit 8-bit handling and operations on multibyte data words. In particular, microprogrammed minicomputers may be furnished with instruction sets especially adapted to those of an associated large digital computer [24].

Unfortunately, today, many of the so-called front end processors involving minicomputers are only direct lower cost replacements for the traditional line adapter units and do nothing to relieve the main processor congestion caused by the communications processing requirements. Even though this might be considered a step in the right direction, it does not go far toward solving the real communications cost problem. Studies have shown that a significant cost saving is readily available if the communications processing is allocated to the lower cost minicomputer [25].

Even when only one large data processor is used, the application of a minicomputer communications front end processor significantly simplifies the communications function. By isolating the communications processing

function, a more realistic system evolves, allowing better network control, easier modification for expansion, flexibility to adapt to changes in system requirements and independence from the major tasks of the data processor.

3. Desirable Features in the Minicomputer Front End

Keeping the above in mind, what then are the generally desirable features in such a minicomputer, and how should it be interfaced?

The most important requirement is, of course, the ability to interface the minicomputer to the mainframe CPU. The interface hardware normally appears to the minicomputer as one of its peripherals and to the mainframe CPU as a device on either a selector or multiplexer channel. It is usual for the interface to be programmable, from the minicomputer, to appear to the mainframe CPU like any of its peripherals.

The architecture of the minicomputer itself is not critical. It is desirable that instructions should be available to facilitate the manipulation of characters, because that is, of course, the main function of a front-end processor. The ability to manipulate queues, stacks and lists is also a useful feature, however, some architectural characteristics may effect this. The amount of core storage required is unlikely to exceed 64K bytes, because on the average only a small number of terminals will actually be in use, and therefore requiring a buffer, at any instant [26]. However, a direct access device will be necessary to queue messages during overload situations.

The method of introducing the program stored in the front-end processor will affect the required configuration

of the minicomputer. The source program can be loaded from the main computer, or can be punched on to paper tape or cards or encoded onto magnetic tape, in which case a paper tape reader, card reader or magnetic tape reader would be required. This peripheral will not normally be required once the front-end processor is in operation and therefore its cost should preferably be kept low. A preferable method of assembling the front-end processor program is to do it in the mainframe CPU. To do this, one must write, or buy a cross-translator to run in the mainframe CPU. This technique requires some investment in the necessary programs but its advantage is that all the facilities of the mainframe CPU for editing and storage of source programs can be used, as well as fast input peripherals and the line printer [27].

Many minicomputers are available with a mini-executive or operating system. This type of system normally contains a number of peripheral controllers which are useful in implementing a front-end processor [28].

A further facility of the mini-operating system should be the ability to run a number of programs concurrently, with each one protected from corruption by the others [29].

4. The Message Handling Program

The message handling program for a minicomputer front-end is best written as a series of modules. These modules should be small so as to facilitate testing and subsequent enhancements to the front-end processor. The actual functions of these modules depend to some extent upon the overall functions of the front-end processor system and the minicomputer chosen but the following modules are almost

certain to be required [30].

Interrupt handling

The interrupt handling module will be entered when an interrupt occurs, either from the interface hardware to the main machine or from line drivers. This module is quite often provided as part of the minicomputer executive.

Interface driver routine

The interface driver routine module will be entered following an interrupt from the interface hardware and will take data from a queue of messages awaiting transfer to the mainframe CPU. The rate of entry to this module depends upon the design of the interface hardware but it is likely to be entered either once per character transfer or once per message transfer.

Line driver routines

The method of driving lines varies considerably between different minicomputers, so it is difficult to generalize the design of these modules. It has been found advantagous, however, to separate the function of sending one message from the overall functions of handling the full protocol. These modules are entered either once per message, once per character or once per bit. The characters received from each line should be buffered; when a complete message has been received the end of message character will be detected. The contents of the buffer are then added to a queue of messages awaiting further processing by the protocol control modules. It may be necessary to have different line driver routines for different types of lines, again depending on the minicomputer chosen.

Protocol control module

The protocal control module receives complete messages from the line driver modules and decides what action should be taken. This may be to place a message on the queue for transmission to the mainframe CPU, perhaps via an intermediate module, or to analyse the message and to send a subsequent part of it to the terminal, via the line driver module. This module needs to check parity, and in some cases may even need to compute it if no hardware parity check is available.

Intermediate modules

Between the terminal interface and the line driver interface it may be necessary to perform some processing of the data. This could be simple code conversions such as from EBCDIC code to ASCII code, or could be a complex process such as control of a dialogue with an operator at an interactive terminal. The design of these modules depends largely upon the system design.

Buffer pool control

In a front-end system with a large number of lines, it is impossible to allocate one buffer to each terminal or line, so that a system of buffer allocation is required. A good system is to maintain a pool of free buffer blocks, each one with a chain address to the next. A module is then required to provide a block on request or to return a block to the pool.

Calling routine

A module is required to maintain a list of pointers, which point to other modules that need to be called. When

there is no outstanding task, then control should be returned to this routine.

5. Alien Minicomputers as Front-Ends

It is often necessary to choose between the mainframe manufacturer's standard piece of equipment and an alien attachment when choosing a minicomputer front-end.

The mechanics of implementing a front-end system using an alien minicomputer, however, should be approached with caution. As regards the outboard interface, that between the front-end processor and the transmission lines, the following considerations may be relevant. Is there hardware logic available for such functions as line sampling and character assembly? Can the system cope with characters of different length, different transmission speeds, different parity modes, different sets of control characters and so on? Double character buffering and automatic detection of line speed are other features that may be relevant. A prime justification for the use of an alien minicomputer as a front-end processor is often the extra flexibility gained thereby. The disadvantage of such an alien attachment ia that it can lead to problems in the inboard interface [31].

The CPU characteristics of the front-end processor are unlikely to be critical, although a suitable instruction repertoire for byte and bit handling is an advantage. An important point, however, is that the user may be tempted to try to exploit any spare capacity in the front-end CPU [31].

Using spare capacity for purposes that are essentially linked to the communicatons handling may be safe. However, attempts to exploit spare capacity in other

directions can lead to severe problems, particularly if they involve modifying any of the mainframe software; in addition, this goes against one of the original reasons for front-ending: separating a discrete function [32].

E. THE TERMINAL CONTROLLER

Much has been said about the development of intelligent terminals as the applications of teleprocessing continue to expand. The heart of such a terminal is a processor. Often a general purpose minicomputer has been adapted to provide local processing capability, which enhances the capabilities of a terminal when operating in the teleprocessing environment. In essence the minicomputer can be considered as a terminal controller that provides an interface for and controls the functions of a variety of simple terminals such as keyboard/printers or keyboard/CRTs. In some cases it may provide the interface for devices more conventionally thought of as computer peripherals, such as discs, tape transports, line printers or plotters, and provide remote access by these devices to centralized or regional data processing facilities. The terminal controller may also be configured to serve as a data acquisition system interfacing with a variety of card readers, badge readers, event sensors, and other special purpose gathering devices [33].

Thus four basic functions are performed by the minicomputer in the terminal controller [33].

- Logical control of the peripheral devices in coordination with the central processor.
- Communication of data to and from the device without error.
- Provision of access to the communications

network in a controlled manner.

- Processing of data that is uniquely required for the particular devices whenever possible.

F. MESSAGE CONCENTRATORS

1. The Advantages of Message Concentration

Message concentrators are designed to reduce the number of long distance communication lines, and they work by accepting data from the low speed terminals and concentrating this traffic on to one higher speed line, for the long distance section of the line transmission. The reverse direction of data flow is handled in the same manner for information flowing from computers to terminals.

Where does the minicomputer fit in this picture? Traditionally, the concentrating function has been performed by Frequency Division or Time Division Multiplexers. However, the plummeting cost of the minicomputer with some simple communications adapters has made it a very powerful tool with which to multiplex communications data. The added flexibility of a programmable concentrator, as opposed to the fixed configuration of a multiplexer, further enhances its usefulness. The way in which the minicomputer concentrator operates allows a higher concentration ratio than the fixed multiplexers. A frequency division multiplexer can make no use of the fact that line utilization is low and allocates a fixed bandwidth to the terminal, whether or not data is being transferred. The programmable concentrator, on the other hand, assembles complete messages, or blocks of characters if the messages are very long, in its memory before making any transmission

on the higher speed line. Consequently, the data rate on the concentrated line is the product of data rate and utilization factor on all the lower speed lines. The result of this feature is that a larger number of lines can be concentrated onto one line of given data capacity, or alternatively, a lower speed concentrated line can be used for a given number of terminals, than is possible with a simple multiplexer [34].

While the cost of a minicomputer-based message concentrator is somewhat higher than that of a multiplexer at present, there are many additional advantages to be gained from adopting this approach.

The introduction of a minicomputer into the communications network at a site remote from the central computer complex allows us to take advantage of its capacity to perform additional functions, which otherwise would have taken up time in the main computer. Any work that can be offloaded from the central computer is usually valuable, since it is the function of the main machine to perform some revenue-producing or cost-saving task. General communications housekeeping merely diverts the machine from its primary work into a non-productive overhead task. The minicomputer, at no additional hardware cost, can perform these tasks and others, such as code conversion, adaptive line speed control, line polling, error detection and correction.

2. Code Conversion

Communications terminals are characterized by their multiplicity of codes, lines speeds and line disciplines. The two most popular codes used for communicating with keyboard terminals today are USASCII and EBCDIC, but many

other codes, specific to an industry or to a user, are to be found, and the older codes are still with us because the equipment using them continues to function adequately.

The central computer performs character manipulation with only one preferred character set, however. At some point in the systerm, code conversion must be performed if the terminal code is different from the native code of the computer. In many communications oriented systems, a variety of terminals will coexist and code conversion is a task specific to each line.

In a basic system, the central computer complex must undertake code conversion, both on input and output for each line, by program reference to look-up tables in memory. If a multiplexer is used in the network, the same is true since the multiplexer is completely transparent to the computer and terminal. All code conversion can be performed simply and the concentrated data flowing in each direction will be in the main computer's native code. In addition, the message arrives from the concentrator as "blocked" information; that is a block of information which originated from one terminal arrives correctly assembled into a message or message segment. In contrast, the data transmitted over individual lines or simple multiplexed lines, is in the form of individual characters arriving at different speeds and they must be assembled into blocks by the main computer. This task, usually performed by a combination of software and hardware scanning, is very time consuming and it is of great benefit to perform this function in the message concentrator.

In a similar manner, other terminal-specific characteristics will be handled completely by the concentrator, and be invisible to the main computer. In this category would be included not only the code and code

length, but also parity generation and checking, deletion and addition of the correct "start" and "stop" bits, detection and generation of special character sequences which have unique interpretations and, of course, maintenance of the correct data speed on each line appropriate to the terminal in use. This last item would, naturally, have to be performed by any communications controller, but adaptive line speed control is yet another area where the minicomputer flexibility is apparent and can provide assistance to the central computer [35].

3. Line Polling

Polling activity can be a significant overhead to the computer. It provides one method of coordinating traffic flow. The existence of a remote minicomputer concentrator can ease the situation by undertaking the entire polling and selection sequence, thus relieving the control computer to perform more useful work. The polling sequence, when performed by a remote concentrator, can be modified at any time by commands from the central computer so that control is always maintained by the main processor [35].

4. Error Detection and Correction

Another of the many tasks that a communications oriented processor must perform is that of message validation. On today's telephone lines, bursts of noise, clicks, distortion and high background noise are sometimes experienced but are, in most cases, acceptable during voice conversations. However, if a bank is conducting a transaction from a branch office terminal to an account maintained on a central computer file, any error introduced

by a click on the communications line is definitely not acceptable. Elaborate means are used, therefore, to check the accuracy and validity of data before allowing the transaction to proceed.

The particular techniques of error detection are highly application dependent, ranging from visual detection by the operator, in non-critical situations, to the use of redundant codes with an infinitesimal probability of an error passing undetected. Error correction usually is accomplished by re-transmission of the character or message containing the error.

The error detection and correction procedure is terminal-dependent, in the same way as the code procedures described earlier. The computer must, therefore, maintain tables of appropriate procedures for message checking and correction. Again, this is an overhead function that can be un ertaken by the remote minicomputer concentrator, which then passes only completely validated messages to the central computer. The main computer complex now, instead of having to accommodate procedures for every type of terminal that may be connected, has only to detect errors introduced between the remote concentrator and the central site and requires only the one correction procedure [36].

5. Remote Environment

The very fact that the concentrator is remote from the main computer site means that it frequently operates unattended, in a basement or small equipment room for example. Reliability is obviously an absolute requirement and here the minicomputer is outstanding. Its small number of circuits, all integrated and all digital, give a mean time between failures in excess of a year in many cases

[36]. Maintenance, when it is required, is usually rapid due to the small number of circuit boards in the system, and heat dissipation is low so that closely controlled air conditioned environments are not required.

When software changes are needed, for example to improve or alter a class of service performed by the concentrator, it is usual to "down line load" the new program rather than visit each remote site in turn. Down line loading is a special communication mode in which the main computer commands the remote concentrator to consider a message to be an executable program rather than a message to be sent on to a terminal. By this means, the main computer can exercise complete control over all concentrators, the software in each remote location can be updated in minutes instead of days and recovery from power failure at the remote location is easily accomplished. An extension to the usefulness of down line loading is remote diagnosis of faults. Here, if a hardware failure at the remote site is reported to the central computer, a diagnostic program can be down line loaded and the results communicated back to the central site. In this way, the field maintenance engineer could be informed of the cause of the fault before he leaves his office and can then be certain he has the correct replacement parts with him.

These, then, are the functions of a remote concentrator and the reasons that they are built around minicomputers. As computer-based communications expands, the number of such devices is expected to increase rapidly, due both to their relieving the main computer of significant overhead and to the rapidly falling prices making them competitive with the more traditional multiplexers. Already, large networks are operating sucessfully with unattended minicomputer concentrators as nodal points in the system and are providing a very successful tool in reducing

operational costs [36].

G. COST EFFECTIVENESS COMPARISONS: HARDWIRED VS. MINI VS. MICRO

1. The Need for Cost Comparisons

Despite the enthusiasm that can be generated in behalf of applying mini/microcomputers to data communication functions- from the terminal to the front end processor- the present roles of these machines must be placed in perspective. They are not always cost effective. Pragmatically, the minicomputer and microcomputer are simply ways to implement the numerous functions that must be carried out on a data communication link. The other alternative is the hardwired device.

The three approches will be examined here for cost effectiveness in carrying out data communication functions.

2. Advantages and Disadvantages of Each Device

Hardwired devices appeared first. Their main characteristic is lack of programability. That fact restricts the versatility of the device for the user. Hardwired devices can range from electromechanical relays to devices composed of many discrete components (diodes, transistors and resistors) and thus can be unreliable, inflexible, and expensive by today's standards.

The stored program minicomputer and microcomputer have been described earlier.

Table One [37] gives a comparison of several mini and micro computers. Note that, on the average, the execution time for the first seven instructions is about two to four times slower for the microcomputer then for the minicomputers. Note also that the multiply and divide instruction execution times are smaller in the minicomputers which have multiply and divide hardware. The microcomputer in this example uses subroutines.

Generally, the speed of a microcomputer is limited by the speed of the microprocessor. The Intel 8080 package is an example of one of the average processors available today. It has a 2 microsecond cycle time 8-bit microprocessor.

To give and idea of cost, each Intel 8080 sells today for about $29.95, with the cost expected to drop to $15 within a year [38].

3. <u>A Cost Effective Comparison: The Line Interface Unit</u>

Consider, as a simple minded example, a line interface unit (LIU), that can be used either with a remote terminal or a front end processor. What may be involved is the implementation of a binary synchronous communications data link control, including a cyclic redundancy check for error control, and code conversion. The cyclic redundancy check (CRC) is the most time consuming task. As shown in table two [39], a microprocessor built around the Intel 8080 microprocessor takes about 370 microseconds, on the average, to perform a CRC-16 algorithm [37].

Suppose the line operates at 9,600 bits per second. That is, suppose 1200 8-bit characters appear on the line each second. This means the CRC operation occupies 1,200 x

0.370 milliseconds or 444 milliseconds each second or 44.4 percent of the avaliable CPU time.

Suppose, also, that each incoming character is in Baudot code and must be converted to ASCII code with parity. As Table Two shows, such a code conversion takes between 15 and 25 microseconds. Using an average of 20 microseconds, converting 1,200 8-bit characters each second takes 24 milliseconds, or about 2.4 percent of the available time.

In short, a line interface unit employing a microcomputer, for the situation described above, can perform all its functions and still have processing time left over. That is, at 9,600 bits per second the LIU is not throughput limited. In fact, such a LIU could handle 19,200 bits per second transmission and still be marginally below saturation. With a slight change in the software, such a line interface unit could handle up to sixteen 1,200 bit per second lines. However, it must be kept in mind that operating near saturation will cause the queue to become quite large. Although this example is simple minded, because a hardwired device would normally be used, it illustrates what can be done with software.

Table Two shows, that despite the versatility of the microcomputer based line interface unit, the minicomputer can do the job faster, and probably at less cost per message. Then why consider the microcomputer at all? The answer depends, for one thing, on whether the line interface functions are to take place at the terminal or at a minicomputer based front end processor.

If throughput is relatively small, and enough unused memory is available, it may be more effective to have such interface operations as data link control, error control, and code conversion performed by the minicomputer. There

would still be little economic penalty for handling the additional load. But if the minicomputer is nearing saturation, and its processor must handle more throughput because of a network expansion, then the microcomputer, in the form of a self-standing line interface unit as a low-cost preprocessor for the minicomputer, tends to be the logical choice [37].

However, a microcomputer used as an LIU at a terminal is an obvious choice over a minicomputer. At a terminal only certain prescribed functions need to be accomplished, such as error control, line protocol, and perhaps code conversion. In a case like this, a minicomputer's capability is so great that using it would be a case of overkill, and a waste of money.

4. Other Cost Comparisons

As already mentioned, the three alternatives to implementing data communications functions (hardwired device, minicomputer, microcomputer) will each yield a different cost effectiveness. Table Three [40] lists 12 common data communications functions. Each alternative is rated for cost effectiveness, assuming that just the indicated function is provided by the equipment. It must be noted that this is strictly the opinion of one author.

Note that certain functions can not be done at all by hardwired devices, as indicated by a zero in the table matrix. On the other hand, the table shows that implementing CRC error control is very cost effective (three stars) for the hardwired device, not so cost effective (one star) if that is all the microcomputer has to do, and a definite case of overkill (open square) if that is all the expensive minicomputer has to do. The table shows that the

microprocessor is very cost effective (three stars) or cost effective (two stars) for such functions as data compression, buffering, buffer management, format control, and text editing. As we can see, the microcomputer proves very beneficial as a processor in advanced programable terminals [39]. Table Three also shows whether a function can be located at a terminal, a front end processor, or both.

Such functions as code conversion, error control, and handshaking when taking place at the terminal's line interface can also be accomplished in an overall cost effective manner with microcomputers.

Another look at the data in Table Three reveals that the minicomputer is an expensive device to serve simply as a line interface for a terminal. However, when the minicomputer is intended to implement all the functions of a front end processor, it can be very cost effective.

However, should a front end processor approach saturation, the next step would be to remove certain low speed functions - CRC calculation, for example - from the minicomputer and embed them in a line interface unit (a microcomputer) serving as a preprocessor.

5. An Example of Dual Importance of Microcomputers and Hardwired Devices

One area of data communications which shows the dual importance of microcomputers and hardwired devices is that of error control. Let us consider that area in more detail.

As mentioned earlier, the defined microcomputer can handle a 19,200 bits per second line, but in doing so it

would use up almost all available processing time. Consider, then, what would happen if the line's transmission rate was 56,000 bits per second: the microcomputer would run out of time (saturate) and could not service the redundancy check for each data block (at 370 microseconds per CRC calculation; the microcomputer would saturate at 21,600 bits per second) [41].

Hardwired devices offer a solution in such a situation. Instead of computing each cyclical redundancy check, an alternative is to add to the microcomputer a special shift register, designed specifically for CRC; Motorola's MC 8503 universal polynomial generator is typical. It is a nonprogramable integrated circuit package. Here, the hardwired redundancy check is started using only one instruction from the microcomputer's control memory to the CRC generator. But the arithmetic logic unit (ALU) itself is free to go on to other tasks after issuing its START instruction. If the total ALU START interval for a redundancy check is, say, five microseconds, then the error checking of the 7,000 8-bit data characters arriving each second at the line interface unit will take less then 5 percent of the LIU's throughput capability [41].

Adding the polynomial generator to the microcomputer does raise the microcomputer's cost, but it frees about 40 percent of the LIU throughput capacity for other chores [40].

VI. MINI AND MICRO COMPUTERS IN NETWORKS

A. THE ADVANTAGES OF NETWORKS

The most common configuration of computers with remote terminal attachments has a communication arrangement controlled by one central machine with the lines and terminals connected radially from it. However, the full possibilities afforded by the marriage of computers and communication network is not realized until a further step is taken. This is the concept of an open-ended, expandable network with no single central control. The prime example of this approach is the telephone network itself which can be expanded by adding elements such as switching offices, trunk lines, local lines, and so on.

The adoption of the network principle allows many computers, often of different types, to perform jobs and to communicate with any terminal on the network. Thus, a terminal operator may have access to a business batch computer, a scientific computer, a time share computer, or to any other remote terminal connected in to the network. Expandability is a key feature of this arrangement. When any element of the network approaches its point of saturation, another element can be added to provide the necessary additional capacity, if this element does not overload other parts of the network. The elements in the network may be geographically located in one room if necessary, or may be distributed across a continent or worldwide, such is the flexibility of the system [42].

The network principle is a very powerful tool in many environments. A distributed data base, for example, can improve communications efficiency by locating the data where it is most frequently used. Assume, for example, a hypothetical airline with coast-to-coast service and a reservation computer in Denver, Colorado. However, suppose 70% of the airline's passengers leaving Boston are going to New York or Washington, D.C. It is not efficient line utilization to transact all the Boston-New York or Boston-Washington, D.C. reservations over a long distance line to Denver. A computer on the East Coast to maintain all the local reservations and connected to the main computer in Denver would mean a significant reduction in the long distance communications traffic. Each computer would have access to the data files of the other for passengers making connecting reservations.

Large corporations with decentralized facilities can also use networks and distributed data bases in a similar manner. File security is crucial in this environment to insure that only qualified users can obtain access to the computers and the data. One of the advantages of network operation is that computing redundancy exists, with the prospect of still being able to perform useful work even when the local processor is temporarily out of service. This is a particularly valuable feature which is further enhanced by the use of a network having several paths through it from source to destination. Thus, even in the presence of line or network processor failures, traffic can still get through, perhaps with some slight additional delay.

The minicomputer plays its usual role in the network environment of handling the data communications checking, routing, formatting, transmitting and converting of messages in a cost-effective manner. It has won for itself an

enviable reputation for reliability when operating unattended 24 hours a day, 365 days a year, and as more users come to appreciate the power of the computing network, more minicomputers will be integrated as critical network processor elements.

B. NETWORK STRUCTURES

Not surprisingly, computer networks have evolved in much the same manner as other types of networks in our society such as those in communications, power and transportation. The evolutionary pattern has been to begin with pairs, which are expanded to chains, which in turn lead to series-parallel arrangements such as trees. The networks are interconnected in a variety of ways. They may be based on superior-subordinate relationships, such as commonly found in task-oriented organizations (e.g., industrial or government networks), or else they may be predicated on a more democratic relationship because of more general goals (e.g., networks for research and education) [43].

Some networks consist of a number of small computers used as an alternative to a single large computer.

A key element in all networks is intercomputer and computer-to-terminal communications. A well established trend is the use of minicomputers as front end processors to larger machines to handle the communications line control functions. In addition to the usual communications processing tasks, the minicomputer may perform a variety of simple data processing tasks on a more cost effective basis than a large machine. The tasks include editing, calculations, and compilation and execution of languages such as BASIC [43]. Although such a simple pairwise

intercommunication could hardly be called a network, there is a trend to use interconnected computers whose functions are specialized. Another similar trend is to use remote job entry computers as intelligent terminals, connected in the form of a star network [43].

There are two basic types of network structures: directly connected networks and store-and-forward networks. The directly-connected structure provides for a direct communications link between the transmitter and receiver. Most of today's networks operate in this manner because it offers the simplicity of a fixed communications pattern. The structure may be a one-level layer or more complicated structure. One example is a star, which is a central site connected to a number of satellites. The characteristics of directly connected networks are listed in Table Four [43].

Store-and-forward networks contain separate switching computers, which provide buffering to insure that the networks operate smoothly regardless of the speeds or data formats of the machines involved. This type of network also permits any machine to communicate with any other in the network. The ARPA network, which is be discussed below, is a good example of this type of structure [44].

C. PACKET SWITCHING NETWORKS: AN EXAMPLE OF MINICOMPUTER USE

The Advanced Research Projects Agency (ARPA) network was conceived in 1968 and placed in operation in 1969 [45].

The purposes of the system are to investigate broadly the uses of a computer network; to explore an alternative method of message switching; to provide a wide range of

computing facilities to a community of computer and physical scientists, for program and file sharing; and to permit the users to communicate with each other in a variety of ways.

The large savings possible from resource sharing are dependent upon having available an economic means of switched communications to access these resources. This communication facility must be reliable, sufficiently responsive to meet all of the interactive demands of time-sharing users, have sufficient data transmission speed for high speed printer and display output, be distributed to scattered users throughout the country, and most important be sufficiently economic so as not to negate the cost savings achieved through resource sharing. In the early 1960's, this kind of communication service was not available. It is only with the extremely rapid development of computers themselves that it has been possible to reduce drastically the cost of providing such a communications service to computer data users.

Since computer input/output typically is extremely bursty in nature, requiring peak data transfer rates many times as great as the average data rate, it is necessary to share among many users a communication channel of fixed data transmission capacity in order to achieve reasonable economy. Without doing this, the cost of data communications would be 10 to 100 times [46] as expensive as the raw communication bandwidth, thereby making resource sharing cost-ineffective over even moderate distances.

Packet-switching technology as used in the ARPANET has been developed to permit the sharing of communication lines by many diverse users. By dividing the traffic into small addressed packets (1000 bits or less in length), extremely efficient sharing of communication resources can be achieved even at the burst rates required by interactive computer

traffic. Since resource-sharing computer usage has only become economically viable with the development of packet switching, it is instructive to look at the cost trends that have made this technology possible. To do this, it is necessary to have a model of the average resources utilized in moving data through a packet-switching network.

The basic design of a packet-switching network, as exemplified by the ARPANET, consists of a collection of geographically dispersed minicomputers called interface message processors (IMPs) interconnected by many 50-kilobit/second (kb/s) leased lines. An IMP accepts traffic from a computer attached to it called a host, formats it into packets, and routes it toward its destination over one of the 50-kb lines tied to that IMP. Each IMP in the network receiving a packet examines the header and, making a new routing decision, passes it on towards its destination, possibly through several intermediate IMPs. Thus, a packet proceeds from IMP to IMP making its way to its destination. The destination IMP collects the packets, reformats them into messages in the proper sequence, and submits them to the destination host computer. Throughout the process, each IMP checks the correctness of the packet by means of both hardware and software based error-control techniques. If the packet is received incorrectly due to transmission error on the line, the IMP does not acknowledge receipt and the preceeding IMP must retransmit the packet, perhaps over a different path. Because the network uses high speed transmission lines and short packets, and all data is stored in high speed primary memory in the IMPs (as opposed to disc drives and other secondary storage devices) average end-to-end transit delay for a packet is 0.1 second [46].

The general design of a packet-switching network has proven to be so successful in the ARPANET that a new

component of the communications industry has been formed - "value-added" packet carriers, offering the public packet communications service on a nationwide regulated basis.

Value Added Networks, (VANs), use minicomputers at node points in the system. Each connects with at least two others. Some minicomputers, called TIPs (Terminal Interface Processors), connect data input devices and host computer systems directly to the network. Other minicomputers are used for switching and data transmission in the network itself. The minicomputers are interconnected by high speed transmission links [47].

Data received by the TIP minicomputers is divided into packets, transmitted to the destination TIP via the high speed links, and combined by that TIP into the original form. This procedure allows the VANs to intermix data from many users on the links between TIPs. VANs charge for the use of the high speed channels by the number of packets transmitted.

The usefulness of VAN service in a particular application depends largely on response time and network tariffs. Response time varies with the data communications equipment, the line control procedures employed, and the loading.

Network operating costs are related most directly to the data transmission capacity. They should be compared to the operating costs of other types of service, such as WATS or multiplexed data transmission, serving an equivalent load at each data source.

VII. SUMMARY

Mini and micro computers are being used in ever increasing numbers to handle the communications processing functions of data communications systems. Some of these functions include message accumulation, line discipline, error detection and correction, message formatting, message addressing and routing, message storing and forwarding, terminal polling, terminal control, data concentration, code conversion, and transmission speed conversion.

The use of mini and micro computers has been necessitated by the increasing complexity of data processing systems in recent years. Today applications involving hundreds of terminals and a wide variety of sources are commonplace. The problem facing the system designer, as increasing numbers of terminals are used with one computer system, is the need to devise means of lowering the overall cost of the network as it continues to grow.

Although the communications processing requirements are normally simple in terms of data manipulation, they are time-critical factors. This characteristic of being time-critical presents a difficult challenge to the large data processing computer. Typically, a large computer system is designed to work best when it can function continuously, executing a full set of program instructions on a given application before branching to another. Even the most modern machines now being produced suffer in efficiency if unscheduled interrupts requiring immediate service occur outside the control and expectation of the processor. This loss of efficiency and increased processing

load has become such a serious problem that new approaches are needed to again develop balanced operations.

A solution to this problem is the use of mini and micro computers as communications controllers. Such devices can perform all the communications overhead functions previously performed by the main computer, thus freeing the main computer to perform its revenue producing or time saving tasks.

As integrated circuit technology reaches even higher levels of sophistication, the speed and capability of mini and micro computers will increase, making them even more valuable in the data communications field.

APPENDIX A

TABLE ONE

Comparison Of Micro and Minicomputer Instruction Execution Times	Intel 8080 Micro-processor (8bit-operation)	Varian V73 Mini-computer (16-bit operation)	Interdata 50 Mini-computer (16-bit operation)
Instruction	Execution Time in Micoseconds		
Sub/Add	2.0	-	1.0
Logical	2.0	-	1.0
Load	3.5	1.32	1.0
Sub/Add (memory ref)	3.5	1.32	3.25
Immediate	3.5	1.32	1.5-3.0
Jump	5.0	1.32	1.5-3.0
Multiply(1)	230(est.)	4.82-5.32	5.5-5.75
Divide(2)	270(est.)	5.15-5.98	9.75-10.25

Notes(1) 8(binary bits) X 16(binary bits)

(2) 16(binary bits) - 8(binary bits)

Ref [37]

APPENDIX B

TABLE TWO

Comparison Of Microcomputer/Minicomputer Function Execution Time	Intel 8080 Microprocessor (8-bit operation)	Interdata 50 Minicomputer (16-bit operation)
Funtion	Execution Time In Microseconds	
Code Conversion (Baudot To ASCII)	15-25	4.5-5.25
CRC-12 Error Check	244-331	12-15
CRC-16 Error Check	311-427	14-17.5

Ref [39]

APPENDIX C

TABLE THREE

Effectiveness Of Implementation Of Data Communications Functions	Implementation Method			Location	
	Minicomputer	Hardwired Devices	Microcomputer		
Function				T	CP
Auto Speed Detection	◻	**	***	-	X
Data Compression	◻	O	**	X	X
Auto Poll	**	*	***	-	X
Buffer(CRT Display)	◻	**	***	X	X
Buffer Management	**	O	**	X	X
Scheduling Task	***	O	*	-	X
Data Link Control	◻	**	*	X	X
Handshaking	◻	**	***	X	X
Error Control(CRC)	◻	***	*	X	X
Code Conversion	◻	***	**	X	X
Format Control	◻	O	***	X	X
Text Editing	**	O	**	X	-

*** Very Cost Effective

** Cost Effective

* Not Cost Effective

◻ Overkill

O Can't Be Done

T Terminal

CP Communications Processor

Ref [40]

APPENDIX D

TABLE FOUR

	Directly Connected
Cost	Cost small for simple networks
Limitations	Not applicable to large nets or long distances. Applicable to limited traffic matrix (e.g. Star)
Ease of design	Simple
Reliability	Based on redundancy
Performance	Fixed performance which can be modified using dial-up to assist in overloads
Applicability Ref [43]	For small minis

LIST OF REFERENCES

1. Vacroux, A. G., " Microcomputers", Scientific American, v. 232, p. 32, May 1975.

2. Vacroux, A. G., " Microcomputers", Scientific American, v. 232, p. 34, May 1975.

3. The Rand Corporation Report R1279, Minicomputers: A Review of Current Technology, Systems andApplications, by D. Hollingworth, p. 23, 1974.

4. L. C. Hobbs and R. A. McLaughlin, " Minicomputers", Datamation, v. 20, p. 50, July 1974.

5. Datapro Research Corporation Report M07-100, Datapro Report on Minicomputers,"What Is a Minicomputer", p. 101, 1973.

6. R. C. Friedman, " Overview of Mini Architecture", Data Management, v. 12, p. 13, February 1974.

7. R. C. Friedman, " Overview of Mini Architecture", Data Management, v. 12, p. 14, February 1974.

8. H. Falk, " A Checkup: Minicomputer Software", IEEE Spectrum, v. 11, p. 52, February 1974.

9. H. Falk, " A Checkup: Minicomputer Software", IEEE Spectrum, v. 11, p. 53, February 1974.

10. H. Falk, " A Checkup: Minicomputer Software", IEEE Spectrum, v. 11, p. 54, February 1974.

11. Datapro Research Corporation Report M07-100-501, Datapro Reports on Minicomputers, "Microcomputers: A

State of the Art Report", p. 501, April 1974.

12. Holt, R. M. and Lemas, M. R., " Current Microcomputer Architecture", Computer Design, v. 13, p. 65, February 1974.

13. Datapro Research Corporation Report M07-100-501, Datapro Reports on Minicomputers, "Microcomputers: A State of the Art Report", p. 505, April 1974.

14. Theis, D. J., " Microprocessor and Microcomputer Survey", Datamation, v. 20, p. 91, December 1974.

15. Theis, D. J., " Microprocessor and Microcomputer Survey", Datamation, v. 20, p. 99, December 1974.

16. Teener, M. and Liles, W., " Microcomputers", Modern Data, v. 8, p. 52-53, February 1975.

17. Teener, M. and Liles, W., " Microcomputers", Modern Data, v. 8, p. 51-52, February 1975.

18. Teener, M., " Minicomputers and Microcomputers", Mini-Micro Systems, v. 1, p. 48, June 1976.

19. J. Martin, Systems Analysis for Data Transmission, p. 5, G. Forsythe, 1972.

20. J. Martin, Systems Analysis for Data Transmission, p. 269, G. Forsythe, 1972.

21. Infotech State of the Art Report 4, Computing Terminals, by Brening, p.240, 1973.

22. Infotech State of the Art Report 4, Computing Terminals, by Brening, p.245, 1973.

23. Infotech State of the Art Report 13, Minicomputers, by Henderson, p. 136, 1973.

24. Korn, G. A., " A Survey of Minicomputer Applications", Computers and Electrical Engineering, v. 2, p.

26-27, 1974.

25. Infotech State of the Art Report 4, Computing Terminals, by Brening, p.247, 1973.

26. Infotech State of the Art Report 13, Minicomputers, by Hughes, p. 427, 1973.

27. Infotech State of the Art Report 13, Minicomputers, by Hughes, p. 427-428, 1973.

28. Infotech State of the Art Report 13, Minicomputers, by Hughes, p. 428, 1973.

29. Infotech State of the Art Report 13, Minicomputers, by Hughes, p. 429, 1973.

30. Infotech State of the Art Report 13, Minicomputers, by Hughes, p. 429-431, 1973.

31. Infotech State of the Art Report 13, Minicomputers, "Applications", p. 138, 1973.

32. Infotech State of the Art Report 13, Minicomputers, "Applications", p. 139, 1973.

33. Infotech State of the Art Report 4, Computing Terminals, by Brening, p.248, 1973.

34. Ball, C. J., " Communications and the Minicomputer", Computer, v. 4, p. 14, September/October, 1971.

35. Ball, C. J., " Communications and the Minicomputer", Computer, v. 4, p. 15, September/October, 1971.

36. Ball, C. J., " Communications and the Minicomputer", Computer, v. 4, p. 16, September/October, 1971.

37. C. J. Riviere and P. J. Nichols, " Microcomputers Unlock the Next Generation", Data Communications, v. 3, p. 25, September/October 1974.

38. R. H. Cushman, " EDN's Second Annual Microprocessor

Directory", EDN, v. 20, p. 47, November, 1975.

39. C. J. Riviere and P. J. Nichols, " Microcomputers Unlock the Next Generation", Data Communications, v. 3, p. 26, September/October 1974.

40. C. J. Riviere and P. J. Nichols, " Microcomputers Unlock the Next Generation", Data Communications, v. 3, p. 28, September/October 1974.

41. C. J. Riviere and P. J. Nichols, " Microcomputers Unlock the Next Generation", Data Communications, v. 3, p. 27, September/October 1974.

42. Ball, C. J., " Communications and the Minicomputer", Computer, v. 4, p. 18, September/October, 1971.

43. C. G. Bell, " More Power By Networking", IEEE Spectrum, v. 11, p. 40, February 1974.

44. C. G. Bell, " More Power By Networking", IEEE Spectrum, v. 11, p. 41, February 1974.

45. C. G. Bell, " More Power By Networking", IEEE Spectrum, v. 11, p. 44, February 1974.

46. L. G. Roberts, " Data By the Packet", IEEE Spectrum, v. 11, p. 48, February 1974.

47. P. D. Moulton, " Value-Added Networks: An Analysis", Modern Data, v. 8, p. 27, November 1975.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 54 Department of Adminstrative Sciences Naval Postgraduate School Monterey, California 93940	1
4.	Professor N. F. Schneidewind, Code 54Ss Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
5.	Professor V. M. Powers, Code 52Pw Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	Commandant (G-PTE-1/72) U. S. Coast Guard Washington D. C. 20591	2
7.	Superintendent Naval Postgraduate School ATTN: LT V.E.Hipkiss, USCG, Code 32 Monterey, California 93940	1

8. Superintendent 1
Naval Postgraduate School
ATTN: LT C. R. Schramm, USCG, Code 32
Monterey, California 93940

9. CDR John T. Vinson, Code 62Vr 1
Department of Electrical Engineering
Naval Postgraduate School
Monterey, California 93940